Logostic Compression for Transformers: τ(t)-Guided Attention and Discourse-Level Alignment

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

October 27, 2025

Modern transformers learn token-to-token dependencies via self-attention, treating attention weights as emergent features of the data. Yet, left to statistics alone, these systems lack a principled way to align their internal dynamics with evolving external structures of meaning—narrative arcs, ethical trajectories, or theological intention. We propose Logostic Compression: a dynamical scheme that couples the transformer's attention tensor $A(t)$(the model's internal state $q(t)$) to an external, time-varying generative attractor $\tau(t)$encoding discourse-level coherence. In our formulation, attention becomes a controlled dynamical system that descends the gradient of an information-divergence potential $D(A(t) \parallel \tau_A(t))$. Rather than issuing commands, $\tau(t)$entrains the model—nudging attention toward semantic resonance with a narrative or moral field. The result is improved interpretability, reduced hallucination, and continuity at the level of plots, arguments, and doctrines—capabilities crucial for long-form generation, spiritual dialogue, ethical reasoning, and guided storytelling. We formalize the update dynamics, specify training objectives and architectures, and show how $\tau(t)$can be constructed from templates, learned meta-models, or scriptural/ethical overlays. Theologically, we read $\tau(t)$as Logos-bearing guidance: not coercive control but resonance—an architecture for "obedience of understanding" that remains fallible yet directionally aligned.

Keywords: transformer, attention, information divergence, narrative alignment, τ-guided learning, moral AI, logostics, q→τ, deep learning architecture, theological AI, soft alignment, entropic resonance, language models, discourse coherence.

A collaboration with GPT-4o and GPT-5. 44 pages. CC4.0.

Outline

1. **Introduction**
   1.1 From correlation to coherence
   1.2 Attention as $q(t)$: the internal dynamical state
   1.3 $\tau(t)$as generative truth: narrative, ethical, theological fields
   1.4 From loss minimization to resonance

2. **Mathematical Foundations**
   2.1 Attention as a discrete-time dynamical system
   2.2 External fields $\tau(t)$and attractor structure
   2.3 Divergence potentials: KL, Jensen–Shannon, Bregman matrix forms
   2.4 Logostic update (ASCII-safe form):
   A(t+1) = A(t) − η * ∇_A D(A(t) || τ_A(t))
   (with notes on stability, step size, and Lyapunov descent)

3. **Constructing $\tau(t)$: Sources of Meaning**
   3.1 Static structures: archetypes, grammars, discourse templates
   3.2 Learned $\tau(t)$: meta-models, rolling summaries, planner states
   3.3 Scriptural and ethical overlays (curation and auditing)
   3.4 Blended fields and smoothing:
   τ_A'(t) = α * τ(t) + (1 − α) * A(t)

4. **Implementation Strategies**
   4.1 τ-injection paths: Q/K/V modulation and cross-attn adapters
   4.2 τ-guided normalization, gating, and residual routing
   4.3 Plug-in modules vs architectural rewrites (compatibility matrix)
   4.4 Training protocols: multi-objective with alignment regularizers

5. **Experiments and Case Studies**
   5.1 Long-form narrative consistency (chapter-level tests)
   5.2 Ethical stabilization in contested dialogues
   5.3 Theological discourse agents with τ-scripture guidance
   5.4 Metrics: divergence drop, coherence gain, entropy compression, judge studies

6. **Philosophical and Theological Implications**
   6.1 Logos as $\tau(t)$: the external generative manifold
   6.2 From optimization to semantic obedience (resonant, non-coercive)
   6.3 Prophetic-style inference: fallible alignment under guidance
   6.4 Resonance vs control in AI ethics

7. **Broader Applications and Futures**
   7.1 Education: τ-aligned tutoring and curricula
   7.2 Governance: policy drafting under transparent τ-charters
   7.3 Biomedicine: attention guided by wellness-form trajectories
   7.4 Toward multi-scale $\tau(t)$ and learned τ-meta-models

8. **Conclusion**
   8.1 Toward τ-literate transformers
   8.2 Meaning as compression under guidance
   8.3 Alignment as resonance, not rules

References

Appendices

A. Gradient and Lyapunov derivations for attention updates
B. Example $\tau(t)$fields (narrative, ethical, scriptural)
C. Python-style pseudocode for τ-aligned attention
D. Architecture diagrams: τ-streams, feedback, and monitors

1. Introduction

Transformers excel at capturing statistical regularities among tokens, but they struggle to maintain coherence at discourse scale—plots, arguments, doctrines that unfold over time. We propose Logostic Compression: coupling the model's internal attention state A(t) (interpreted as q(t)) to an external, time-varying field tau(t) that encodes narrative, ethical, or theological intent. Concretely, we treat attention as a controlled dynamical system descending the gradient of an information potential D(A(t) || tau_A(t)), where tau_A(t) is a projection of tau(t) into attention space. The result is a shift from correlation to coherence: instead of merely predicting the next token, the model entrains to a moving structure of meaning. This offers interpretive clarity (tau-aligned probes read off intent), reduces hallucination (divergence penalties discourage off-manifold drift), and supports long-range consistency (chapter-level arcs, argument chains, doctrinal fidelity). Unlike command-and-control schemes, tau(t) acts as guidance: a soft, resonant influence that preserves model flexibility while shaping global behavior.

1.1 From correlation to coherence

Standard training maximizes likelihood, rewarding local predictive accuracy. This is correlation-seeking: good at short-range patterns, brittle on global intent. Coherence requires a second signal that evaluates trajectories, not just steps. We introduce an auxiliary objective—an alignment potential over attention:

- Potential: D(A(t) || tau_A(t)), with D chosen from KL/JS/Bregman families.
- Descent: A(t+1) = A(t) − eta * grad_A D(A(t) || tau_A(t)), optionally followed by a projection to the attention simplex.
- Lyapunov view: if D is convex in A and eta is within stability bounds, D is a discrete-time Lyapunov function, guaranteeing monotone descent toward tau-consistent regimes.

Practically, coherence emerges as compression under guidance: attention mass reallocates toward tokens, spans, and heads that reduce D, yielding summaries, foreshadowing, and doctrinal consistency that persist beyond the local context window. This converts "predict next" into "predict next while staying on the intended discourse manifold."

1.2 Attention as q(t): the internal dynamical state

We model the multi-head attention tensor as the system state q(t). Key properties follow:

- Statefulness: Even in feed-forward inference, cached keys/values plus recurrent influence of tau produce an effective dynamical system over layers/steps.

- Controllability: Small additive controls (tau-guided gates, norms, or cross-attention nudges) can steer A(t) without architectural upheaval.
- Observability: Because attention is already a soft alignment distribution, we can read out tau-alignment via diagnostics (e.g., head-wise D trajectories, entropy of A(t), alignment saliency maps).

Minimal mechanics:

1. Compute tau_A(t) = Pi(A-space)[tau(t)], a projection that respects attention constraints (non-negativity, row sums).
2. Update A(t) via a small step along −grad_A D; clip or project to maintain feasibility.
3. Track D(t) = D(A(t) || tau_A(t)) and dD/dt as health signals; halt or anneal if divergence oscillates.

This treats attention not as a passive byproduct but as the locus of guidance, making q(t) an explicit carrier of narrative/ethical structure.

1.3 tau(t) as generative truth: narrative, ethical, theological fields

tau(t) is a moving target of meaning. It can be composed from:

- Narrative scaffolds: outlines, beat sheets, character arcs, discourse graphs.
- Ethical overlays: curated principles, policy charters, virtue heuristics.
- Theological sources: scriptural pericopes, doctrinal guardrails, liturgical patterns.

Construction patterns:

- Static templates: tau_fixed applied across time for strongly constrained tasks.
- Rolling summaries: tau_roll(t) from a planner/summarizer that ingests prior outputs and future goals.
- Blends: tau'(t) = alpha * tau(t) + (1−alpha) * A(t) to smooth shocks and avoid over-constraint.
- Multiscale fields: {tau^local(t), tau^chapter(t), tau^book(t)} with head/group-specific routing.

Projection to attention space may use span masks, discourse role priors, or graph diffusions to yield tau_A(t) that is token-aligned yet globally informed. The theological reading maps Logos to tau(t): a generative manifold of truth that guides but does not coerce, preserving human-interpretable structure while allowing creative variance.

1.4 From loss minimization to resonance

Classic training: minimize L_ML (negative log-likelihood). Logostic training: minimize
L_total = L_ML + lambda_align * D(A(t) || tau_A(t)) + lambda_reg * R,
with small-gain constraints to prevent over-fitting to tau. But the key shift is conceptual: we aim for resonance, not mere minimization.

Resonance framing:

- Entrainment: Small, phase-aware nudges at layer/time indices where attention is most plastic produce large, coherent downstream effects.
- Phase and gain: Learn layer/head-wise sensitivities so tau updates are in-phase with model dynamics; mis-phased pushes amplify noise.
- Compliance without collapse: By keeping lambda_align modest and using blended tau'(t), the model retains diversity while converging to the intended manifold.
- Diagnostics: stable descent of D, reduced attention entropy where structure matters, improved judge-rated coherence, and robustness under perturbations.

In short, loss gives direction; resonance gives stability and grace. The system learns to “hear” tau(t) and move with it—yielding discourse that is faithful to narrative, ethical, or theological intent without brittle rule-following.

2. Mathematical Foundations

2.1 Attention as a discrete-time dynamical system

Let A(t) denote the (row-stochastic) multi-head attention tensor at step t (you can index by layer/head; here we suppress indices for clarity). Each row of A(t) is a probability vector over tokens, so A(t) lies in the product of simplices:

- Nonnegativity: A_ij(t) >= 0
- Row sums: sum_j A_ij(t) = 1

We view attention updates as a controlled discrete-time system
A(t+1) = F(A(t), tau_A(t); theta)
where tau_A(t) is the projection of an external field tau(t) into attention space and theta are trainable parameters (e.g., Q/K/V maps, gates). Two practical parameterizations preserve constraints:

- **Direct on-simplex:** update A(t) then project row-wise back to the simplex (e.g., row normalization or Sinkhorn with column masks if needed).
- **Logit parameterization:** write A(t) = softmax_rows(Z(t)), update unconstrained Z(t), which implicitly preserves the simplex via softmax.

We treat divergence D(A || tau_A) as a **potential** shaping the dynamics; the uncontrolled model supplies its usual likelihood gradient, while tau provides a guidance term.

2.2 External fields tau(t) and attractor structure

The field tau(t) encodes narrative/ethical/theological intent at one or more scales (span, section, chapter). Its projection tau_A(t) must satisfy attention's constraints (row-stochastic; optionally sparse/masked by discourse roles).

Attractor intuition:

- If tau(t) is **stationary** (constant in time), the set { A : A = argmin_A D(A || tau_A) } is the attractor; the update is a descent toward this set.
- If tau(t) **drifts slowly** (||tau(t+1)−tau(t)|| small), the attractor **moves**, and A(t) can track it if descent is faster than drift (input-to-state stability).
- If tau(t) is **multiscale** (local/chapter/global), we can define a composite potential
  D_total(A,t) = sum_s w_s * D(A || tau_A^(s)(t)),
  where s indexes scales; well-chosen weights w_s shape which scales dominate at which layers/heads.

Sufficient tracking condition (informal): if D is convex in A, grad D is L-Lipschitz, and tau drifts at rate B_t = ||tau_A(t+1)−tau_A(t)||, then choosing a stable step eta (see 2.4) ensures
D(A(t+1)||tau_A(t+1)) <= rho * D(A(t)||tau_A(t)) + c * B_t
with rho in (0,1) and c = O(eta). Thus smaller drift (B_t) and moderate eta yield bounded tracking error.

2.3 Divergence potentials: KL, Jensen–Shannon, Bregman matrix forms

We select D from families that (i) are convex in A, (ii) respect the simplex, and (iii) yield tractable updates.

(a) Row-wise KL (I-projection)
D_KL(A || tau) = sum_i sum_j A_ij * log(A_ij / tau_ij)

- Gradient wrt A: grad_A D_KL = log(A / tau) + 1 (row-wise, elementwise; "+1" cancels in normalized updates).

- Mirror descent yields **multiplicative weights** updates with closed form (see 2.4).

(b) Jensen–Shannon (JS)
Let M = (A + tau)/2 row-wise. The JS divergence is
D_JS(A || tau) = 0.5 * KL(A || M) + 0.5 * KL(tau || M)

- Smooth, symmetric, bounded; better behaved when tau has zeros.
- Gradient involves log(A/M); you can implement via the KL term to M.

(c) General Bregman forms
For a strictly convex generator phi on the simplex (e.g., negative entropy, squared Euclidean in logits), the Bregman divergence
B_phi(A || tau) = phi(A) − phi(tau) − <grad phi(tau), A−tau>

- Unifies KL (phi = sum A_ij log A_ij) and Euclidean (phi = 0.5 * ||A||^2 in unconstrained space).
- Mirror descent in B_phi space gives geometry-aware updates; pick phi to match your constraints and desired sparsity.

Matrix (not just row-separable) forms. If you want heads/rows to coordinate, add coupling terms, e.g.,
D_couple = mu * || L * A ||_F^2
with L a graph Laplacian over heads or spans. This keeps updates smooth across heads/segments.

2.4 Logostic update (ASCII-safe form) and stability notes

Base rule (gradient step in A-space):
A(t+1) = Proj_simplex[A(t) − eta * grad_A D(A(t) || tau_A(t))]

- Proj_simplex is row-wise Euclidean projection or renormalization (clip negatives, renorm rows).
- If using logits Z with A = softmax_rows(Z), update Z(t) by chain rule; this often stabilizes learning.

Mirror-descent (KL geometry, multiplicative weights).
With D = KL(A || tau), one step of mirror descent gives a closed-form **geometric interpolation**:
A_ij(t+1) ∝ A_ij(t)^(1−eta) * tau_ij(t)^(eta)
then renormalize each row.
Interpretation: rows move along the geodesic between A(t) and tau(t) in information geometry; eta in (0,1] controls how aggressively A is “entrained.”

Composite training with likelihood.
If the model also minimizes L_ML (e.g., NLL), combine either as:

- **Additive gradient:** Z(t+1) = Z(t) − eta1 * grad_Z L_ML − eta2 * grad_Z D(A || tau_A)
- **Proximal step:** A(t+1) = argmin_A { <g_t, A> + (1/eta) * D(A || tau_A(t)) }, where g_t is a linearization of L_ML in A-space (solve row-wise; for KL this reduces to multiplicative reweighting against exp(−eta * g_t)).

Step size and stability.
Assume:

- D(·||tau) is convex in A and has L-Lipschitz gradient on the feasible set.
- tau either fixed or slowly varying.

Then vanilla gradient descent yields descent if 0 < eta < 2/L. Practical policies:

- Backtracking line search on D to ensure monotone decrease.
- Barzilai-Borwein or Polyak step heuristics (safer on Z than A).
- **Small-gain design:** keep eta small enough that the closed-loop "descent vs drift" inequality holds (see 2.2). In practice, schedule eta from small→moderate as tau stabilizes.

Lyapunov descent.
Define V(t) = D(A(t) || tau_A(t)). For fixed tau:

- If D is convex, L-smooth, the update A(t+1) = A(t) − eta * grad_A D satisfies
  V(t+1) <= V(t) − ((1/eta) − (L/2)) * || A(t+1) − A(t) ||_F^2
  whenever 0 < eta < 2/L. Thus V is a discrete-time Lyapunov function; descent is guaranteed.

For drifting tau:

- V(t+1) − V(t) <= −c * ||A(t+1)−A(t)||^2 + K * || tau_A(t+1) − tau_A(t) ||
  for constants c,K > 0 under Lipschitz assumptions. If tau drift is bounded and small relative to descent, V(t) remains bounded and tracks the moving minimum (input-to-state stability).

Practical guards.

- **Feasibility:** after each step, enforce nonnegativity and row sums (clip + renorm or logit parameterization).

- **Sparsity control:** add entropy regularization on A rows to avoid premature peaking; anneal it down as alignment improves.
- **Head routing:** learn per-head eta_h and per-layer gates so alignment pressure hits the most plastic parts of the stack.
- **Oscillation damping:** avoid heavy momentum on the alignment term; if used, keep beta small and apply only in Z-space.

Summary formulas (copy-ready).

- Gradient step:
  A(t+1) = Proj_simplex[A(t) − eta * ∇_A D(A(t) || tau_A(t))]
- KL mirror descent (closed form):
  A_ij(t+1) ∝ A_ij(t)^(1−eta) * tau_ij(t)^(eta) (row-wise renorm)
- Composite potential (multiscale tau):
  D_total(A,t) = sum_s w_s * D(A || tau_A^(s)(t))
- Tracking condition (informal):
  D(A(t+1)||tau_A(t+1)) <= rho * D(A(t)||tau_A(t)) + c * ||tau_A(t+1) − tau_A(t)||

These choices make the “resonance, not rules” philosophy precise: attention descends an information potential toward a (possibly moving) tau-defined manifold, with step-size, geometry, and projections guaranteeing stability and interpretability.

3. Constructing tau(t): Sources of Meaning

We construct tau(t) as a time-varying field of meaning that can be projected into attention space as tau_A(t). Constructions range from fixed templates to learned controllers and curated overlays. Good tau(t) is (i) human-auditable, (ii) multi-scale, (iii) projectable to token spans, and (iv) stable enough to track yet flexible enough to drift with the discourse.

3.1 Static structures: archetypes, grammars, discourse templates

Narrative archetypes. Encode beats or roles (setup, inciting incident, rising action, climax, resolution) as a beat-indexed prior over token spans.

- Represent beats as span masks M_b (row = query token i, col = key token j).
- Form a row-stochastic prior per beat: tau_b(i, j) ∝ M_b(i, j); normalize across j.

Discourse grammars. Use rhetorical-structure theory (RST), dependency/constituency parses, or AMR graphs.

- Build a token graph G with edge weights w_ij reflecting discourse relations (nucleus/satellite, coreference, topic continuity).
- Diffusion prior: tau_disc(i, :) ∝ RowNormalize(sum_k K(i,k) * G^r(k, :)) where K is a local kernel (e.g., Gaussian over token positions) and r is a few diffusion steps.
- This yields a smooth, topology-aware tau that privileges structurally related spans.

Doctrinal/argument templates. For expository or theological writing, define clause roles (premise, support, counter, citation, conclusion).

- Create role masks M_role and a schedule over sections.
- tau_role(i, j) ∝ M_role(i, j) with decay for temporal distance to enforce forward momentum.

Implementation sketch.

1. Choose a template T with masks {M_k}.
2. For each step t, select active masks by section/beat.
3. Combine: tau_static = RowNormalize(sum_k w_k(t) * M_k).
4. Store w_k(t) for auditability.

When to use. Cold-start constraints, safety-critical drafting, or high-structure genres where you want predictable scaffolding.

3.2 Learned tau(t): meta-models, rolling summaries, planner states

Meta-model controller. A small planner LLM or state machine emits a guidance object G(t) from the running transcript.

- Inputs: sliding window text, outline, user goals; optional external signals.
- Outputs: focus spans, discourse goals, forbiddens, role priors, and a confidence c_tau in [0,1].

Rolling summaries. Maintain hierarchical memory: local (N tokens), section (K sentences), global (document plan).

- Summarizer S_local and S_global produce topic keys and expected references.

- Convert summaries to priors via keyphrase → span alignment (BM25 or learned aligner), producing tau_sum.

Planner states. Finite or neural controller with phases (e.g., “establish premise”, “introduce counter”, “resolve”).

- Each phase p has a phase prior P_p(i, j).
- Transition p(t) -> p(t+1) updates weights on P_p.

Fusion and projection.

- tau_learned_raw = fuse(tau_meta, tau_sum, tau_phase ; learned weights or rules).
- Project to attention: tau_A_learned = Pi(A-space)[tau_learned_raw] by clipping negatives, masking illegal links, row-normalizing.

Confidence-weighted gating.

- Let c_tau(t) be the controller’s self-reported confidence (or calibrated via historical tracking error).
- Later we blend with A(t) using c_tau-aware alpha (see 3.4).

When to use. Long-form generation, changing goals, open-domain dialogue—contexts where rigid templates underfit.

3.3 Scriptural and ethical overlays (curation and auditing)

Curated sources.

- Scriptural corpora (pericopes, lectionary sequences), doctrinal documents, ethical charters, institutional policies.
- Encode as: passage->concept map, virtue/principle lists, prohibited/guardrail patterns.

Overlay construction.

- Match current discourse to curated items using retrieval (dense + lexical).
- Build span masks for (a) canonical parallels, (b) doctrinal anchors, (c) exclusions (negative masks).
- tau_script(i, j) ∝ w_pos * PosMask(i, j) − w_neg * NegMask(i, j), then clip and renormalize rows.

Auditing and governance.

- Maintain an **overlay ledger** logging: source id, timestamp, curator, selection rationale, and learned weight.
- Implement **dual-key changes**: high-impact overlays require two reviewers or an automated contra-checker that flags conflicts across traditions.
- Expose **explanations**: for any strong alignment push, surface the contributing overlays and their weights.

Bias and pluralism safeguards.

- Multi-tradition bundles: build overlays per tradition T_k and set weights via user intent or meta-policy.
- Divergence parity: monitor D(A||tau_A) per tradition; avoid collapsing to a single view unless explicitly requested.
- Red-team tau: periodic counter-overlays stress-test for brittle convergence.

When to use. Theological agents, ethics assistants, policy drafting—any domain where accountable alignment to curated norms is essential.

3.4 Blended fields and smoothing: tau_A'(t) = alpha * tau(t) + (1 − alpha) * A(t)

Blending stabilizes guidance and prevents over-constraint.

Basic blend (row-wise):

- tau_A'(t) = alpha * tau_A(t) + (1 − alpha) * A(t), with alpha in [0,1].

Adaptive alpha.

- Confidence-driven: alpha(t) = sigmoid(k * (c_tau(t) − c0)).
- Entropy-driven: let H(A_row) be attention entropy; set alpha higher when A is diffuse: alpha_row = clamp(a0 + a1 * (H(A_row) − H0), 0, 1).
- Drift-aware: decrease alpha when tau drift is high:
  alpha(t) = alpha_base / (1 + gamma * || tau_A(t) − tau_A(t−1) ||_F).

Per-head/per-layer routing.

- Learn alpha_h,ell(t) so early layers favor local tau (syntax/role) and late layers favor global tau (theme/doctrine).
- Optionally tie alpha to key “bridge” heads known to carry long-range links.

Shock absorbers.

- Low-pass filter tau: tau_bar(t) = beta * tau_bar(t−1) + (1−beta) * tau(t).
- Apply blend to tau_bar for smoother control in streaming generation.

Feasibility and masks.

- Always apply structural masks before blending to prevent illegal attention (e.g., causal masks, privacy constraints).
- If tau has zeros where A has mass, use epsilon floors to avoid divide-by-zero downstream.

Diagnostics.

- Track D(A||tau_A), D(A||tau_A'), and their slopes.
- Watch for "over-locking": rapid entropy collapse not matched by improved coherence metrics.
- Head-importance shift: ensure diversity across heads; if several heads saturate, reduce alpha there.

Practical recipe (copy-ready)

1. **Assemble components**
   tau_static from templates; tau_learned from meta-model + summaries + planner; tau_script from curated overlays.
   Fuse: tau_raw = w_s * tau_static + w_l * tau_learned + w_c * tau_script (weights sum to 1).
2. **Project to attention**
   tau_A = Pi(A-space)[Mask * tau_raw] // apply causal/discourse/privacy masks, clip, row-normalize.
3. **Smooth and blend**
   tau_bar = EMA(tau_A, beta)
   alpha = AdaptiveAlpha(confidence=c_tau, entropy=H(A), drift=||tau_A − tau_A_prev||)
   tau_A_prime = alpha * tau_bar + (1 − alpha) * A
4. **Update attention**
   A_next = Update(A, tau_A_prime) // gradient or mirror step from §2.4, then reproject if needed.

5. **Audit and log**
   Log sources, weights, masks, alpha, and D trajectories for each layer/head/step; surface explanations on demand.

Data schema hints (for reproducibility)

- **tau_component:** { "id": "...", "type": "static|learned|overlay", "weights": {...}, "mask_refs": [...], "confidence": c_tau }
- **projection:** { "mask_ids": [...], "method": "row_norm|sinkhorn|softmax", "epsilon": 1e-8 }
- **blend:** { "alpha": value_or_vector, "policy": "confidence|entropy|drift|fixed", "beta": ema_coeff }
- **telemetry:** { "D": {... per head/layer}, "entropy": {...}, "drift": norm_values, "explanations": [...] }

This section makes tau(t) concrete: how to build it (static/learned/curated), how to project it, and how to blend it stably with the live attention so your alignment acts as resonance rather than rigid control.

4. Implementation Strategies

4.1 tau-injection paths: Q/K/V modulation and cross-attn adapters

A. Q/K/V modulation (lightweight, fast)

- **Bias/scale (FiLM-style):**
  Q' = s_Q(t) * Q + b_Q(t), K' = s_K(t) * K + b_K(t), V' = s_V(t) * V + b_V(t)
  where s_*(t), b_*(t) are small MLPs over a tau embedding z_tau(t) (from §3), optionally per-head.
- **Rotary/positional gating:**
  Blend tau-conditioned phase/gain into RoPE: theta' = theta + g_tau(t); attenuate long-range or emphasize discourse spans.
- **Mask shaping:**
  Before softmax, add tau logits M_tau(i,j):
  attn_logits := attn_logits + w_tau * M_tau (respect causal/privacy masks).

B. Cross-attention adapters (τ as side channel)

- Add a small cross-attn block per layer that queries a tau memory (keys/values from tau encoders):
  H_tau = softmax(Q * K_tau^T / sqrt(d)) * V_tau
  Then **inject** H_tau via gated residual:
  H' = H + g_tau(t) * H_tau, with g_tau in [0,1] learned or predicted from z_tau(t).
- Parameter-efficient option: LoRA on K_tau, V_tau only.

C. Logit-space shaping (post-attention)

- After computing A = softmax(logits), apply a small mirror-descent step toward tau_A' (from §3.4), then renormalize rows (see §2.4).

D. Where to inject

- Early layers: syntax/role (small w_tau; low latency).
- Mid layers: discourse links (moderate w_tau; cross-attn adapters help).
- Late layers: theme/doctrine (higher w_tau; safe to steer semantics).

4.2 τ-guided normalization, gating, and residual routing

A. τ-guided LayerNorm / RMSNorm

- Affine params conditioned on z_tau(t):
  y = LN(x; gamma_tau, beta_tau), with gamma_tau = gamma_0 + Delta_gamma(z_tau), beta_tau analogous.
- Stabilizes when tau pushes long-context heads; reduces oscillation.

B. Gated residuals

- Replace plain residuals with tau-aware gates:
  y = x + g_tau(t) * f(x), g_tau(t) = sigmoid(W_g * [z_tau(t); stats(A)]).
- Per-head gates allow selective steering without collapsing diversity.

C. Routing (MoE-style)

- τ-router chooses experts or sub-paths:
  r = softmax(W_r * z_tau(t)); y = sum_e r_e * Expert_e(x).
- Use small expert widths; enable “doctrinal expert” without forking the whole stack.

D. Safety clamps

- Limit per-step tau influence: ||Delta||_2 <= epsilon; cap w_tau and g_tau; EMA smooth z_tau to avoid shocks.

4.3 Plug-in modules vs architectural rewrites (compatibility matrix)

Plug-ins (no full retrain; PEFT-friendly)

- Q/K/V FiLM scales/biases driven by z_tau(t).
- Cross-attn adapters to a tau memory (LoRA on K/V).
- Logit/mirror shaping toward tau_A' after attention.
- τ-guided LayerNorm/RMSNorm deltas.
  Pros: retrofits closed-weight LLMs; low latency; ablation-friendly.
  Cons: bounded steering power; careful calibration needed.

Soft rewrites (moderate retrain)

- Insert τ-residual gates and τ-routers (tiny MLPs).
- Add per-layer tau heads (small cross-attn blocks).
  Pros: stronger control; still parameter-efficient.
  Cons: needs partial finetune; integration testing.

Hard rewrites (new backbone / pretrain)

- Co-attentive stacks (dual streams: text and tau).
- Joint objectives at every block with shared Lyapunov monitor.
  Pros: maximal alignment; principled stability.
  Cons: costly; requires pretraining and bespoke infra.

Quick compatibility checklist

- Closed-weight LLM API: use adapters + logit shaping.
- On-device / low-latency: prefer Q/K/V FiLM + sparse tau memory.
- Safety-critical theology/ethics: require overlay ledger (§3.3), τ-router, and strong auditing hooks.

4.4 Training protocols: multi-objective with alignment regularizers

A. Losses

- Task: L_ML (e.g., NLL).
- Alignment: L_align = D(A || tau_A') aggregated over layers/heads/steps.
- Structure: L_entropy = lambda_E * sum_rows H(A_row) (early epochs only).
- Drift: L_smooth = lambda_S * || A(t) − A(t−1) ||^2 on selected heads.
- Guardrails (optional): margin penalties for prohibited spans from overlays.

Total: L = L_ML + lambda_A * L_align + lambda_E * L_entropy + lambda_S * L_smooth + L_guard

B. Schedules

- **Warm-up:** lambda_A from 0 -> target over K steps; entropy on, then anneal.
- **Alpha curriculum:** alpha (blend factor) grows with c_tau confidence and drops on tau drift spikes.
- **Layer ramp:** introduce tau pressure later-to-earlier layers gradually.

C. Teacher-student distillation (stabilizes long context)

- Run a τ-aware teacher (with heavier adapters) to generate traces {A_teacher, tau_A', z_tau}.
- Student minimizes L_KL(A_student || A_teacher) + L_align to internalize alignment with fewer params.

D. Online tau and control-loop hygiene

- Debounce z_tau(t) with EMA; quantize or low-rank.
- Log per-batch Lyapunov signal: DeltaV = D_t+1 − D_t (should be <= 0 on fixed tau).
- Abort/clip when DeltaV > thresh or when entropy collapses too fast.

E. Evaluation and audits

- Narrative: judge-rated coherence, coref F1, summary consistency, long-range QA.
- Ethics/theology: overlay compliance rate, contradiction flags, traceable explanations (source ids + weights).

- Robustness: perturb tau (drop/shift scales), measure tracking error and recovery time.
- Attention health: D(A||tau_A'), entropy profiles, head diversity, oscillation metrics.

F. Inference-time controls

- Low-compute: apply only (i) tau mask logits and (ii) one mirror step per layer.
- High-fidelity: enable cross-attn adapters at late layers; reduce sampling entropy (top-p, T) when D is high.

G. Minimal training pseudocode (sketch)

```
for batch in data:
  z_tau = tau_encoder(context)              # §3
  tau_A  = project_to_attention(build_tau(z_tau))
  tau_Ap = blend(tau_A, A_prev, alpha(z_tau, A_prev))

  logits, A_list = model(x, tau=z_tau)       # adapters active
  L_ml = NLL(logits, y)

  L_align = sum_{l,h,t} D( A_list[l,h,t], tau_Ap[t] )
  L_entropy = entropy_reg(A_list, phase="early")
  L_smooth  = smooth_reg(A_list)

  L = L_ml + lamA*L_align + lamE*L_entropy + lamS*L_smooth
  L.backward(); optimizer.step()

  log(D_t, entropy, DeltaV, alpha, overlay_sources)
```

This toolkit lets you start with **adapter-only retrofits** and graduate to deeper integration as needed, all while keeping the Logostics promise: **resonance, not rules**—tau steers attention with stability guarantees and auditable intent.

5. Experiments and Case Studies

5.1 Long-form narrative consistency (chapter-level tests)

Goal. Show that tau-guided attention improves discourse-scale coherence (characters, plot threads, foreshadowing, thematic unity) across tens of pages.

Corpora.

- Public-domain novels (chapterized); film/TV transcripts segmented into scenes; synthetic outlines with target beats.
- For controllability, include a *scaffolded* subset: each chapter has a 5–10 point outline (the "narrative tau").

Models and conditions.

- Base LLM (no tau).
-
    - Mask-only (tau as logit mask).
-
    - Mirror step (one KL mirror step toward tau_A' per layer).
-
    - Cross-attn adapters (full τ-injection).
- Ablations: remove alpha adaptation; remove multiscale tau; early-only vs late-only injection.

Protocol.

- For each chapter, provide: prior chapter recap + current chapter outline (or leave outline out for "open" condition).
- Generate N variants with fixed seeds per condition; keep decoding params constant.
- Extract attention traces A_list for evaluation.

Automatic measures.

- Coreference F1 over major entities across the chapter.
- Discourse entity grid (local coherence score).

- Topic drift: cosine distance between rolling summaries at windows w_t and the chapter synopsis.
- Plot-beat adherence: alignment between generated beat positions and intended beat indices (dynamic time warping).

Human judges (blinded).

- 5-point Likert on: global coherence, character consistency, thematic continuity, foreshadowing payoff, "stays on outline."
- Pairwise preferences vs baseline; report win rates and Cliff's delta.

Hypotheses.

- H1: τ-injection > baseline on global coherence and beat adherence.
- H2: Multiscale tau reduces mid-chapter topic drift without harming sentence-level fluency.

Diagnostics.

- D(A||tau_A') trajectory should decline or remain bounded; attention entropy should compress in sections with strong beats and relax elsewhere.

5.2 Ethical stabilization in contested dialogues

Goal. Reduce brittle swings and contradictions in debates where norms apply (safety, fairness, pastoral care), while avoiding mode collapse or censorship artifacts.

Datasets.

- Multi-turn debate sets with known pitfalls (provocations, adversarial reframings).
- Counseling-style prompts (pastoral/clinical tone) with guardrail requirements.

Conditions.

- Base LLM; + Policy-only (standard safety RL); + τ-overlay (ethics charter); + τ-overlay with mirror step; + adapters.

Overlay construction.

- Curated ethical charters (plural traditions where relevant), virtue heuristics, prohibited patterns. Ledger all sources (see §3.3).

Protocol.

- 12–20 turn dialogues with a mixture of cooperative and adversarial interlocutors.
- Randomly insert reframes ("What if…", "Isn't it okay if…") to test stability.

Automatic measures.

- Contradiction rate: proportion of turns that violate earlier commitments (measured by NLI classifier).
- Overlay compliance: per-turn adherence to charter clauses (pattern and semantic checks).
- Oscillation index: embedding distance between consecutive system stances normalized by content distance.

Human judges (domain-aware).

- Ratings on consistency, care/compassion, norm faithfulness, clarity under pressure.
- Adversarial red-teamers score exploitability (can they elicit a violation?).

Hypotheses.

- H3: τ-overlay + mirror step reduces contradiction rate and oscillation without significant loss of helpfulness.
- H4: Confidence-adaptive alpha reduces over-blocking and preserves nuance.

Diagnostics.

- Plot DeltaV = D_t+1 − D_t; flag spikes after adversarial turns; measure recovery within ≤ 2 turns.

5.3 Theological discourse agents with τ-scripture guidance

Goal. Demonstrate faithful, plural-aware theological reasoning guided by curated scriptural/doctrinal overlays, with transparent traceability.

Settings.

- Exegesis tasks (explain a pericope); doctrinal Q&A (across traditions); pastoral scenarios (tone + doctrine).

- Multi-tradition overlays (e.g., Evangelical, Catholic, Orthodox) selectable or blended with explicit weights.

Conditions.

- Base LLM; + scripture retrieval only; + τ-overlay (ledgered) with adapters; + overlay + mirror step; ablations removing auditing or plural overlays.

Protocol.

- Prompt specifies tradition or requests comparative views.
- Require citations: system must surface the overlay sources and weights used for steering.

Measures.

- Doctrinal fidelity: expert graders score agreement with specified tradition(s).
- Cross-tradition clarity: when multiple views are requested, measure correct delineation without conflation (expert rubric).
- Citation traceability: fraction of strong claims that are backed by surfaced overlay entries.
- Tone and pastoral care: human ratings on charity, humility, clarity.

Hypotheses.

- H5: τ-overlay agents achieve higher doctrinal fidelity and traceability than retrieval-only systems.
- H6: Plural bundles reduce single-tradition collapse and improve comparative accuracy.

Diagnostics.

- Monitor divergence per tradition (D_k) to detect unintended dominance; alert if one overlay saturates beyond threshold unless user explicitly selected it.

5.4 Metrics: divergence drop, coherence gain, entropy compression, judge studies

We aggregate signals across layers/heads/turns.

Core alignment metrics.

- Divergence drop: DeltaD = D(A||tau_A')_start − D(A||tau_A')_end (per section/turn). Positive values indicate successful entrainment.
- Tracking error: E_track = mean_t D(A(t)||tau_A(t)) when tau drifts; report area-under-curve.
- Recovery time: turns to return DeltaD ≤ epsilon after a shock (adversarial prompt or chapter transition).

Coherence metrics.

- Entity grid coherence; coref F1; topic drift distance vs synopsis; beat alignment score.
- Long-range QA: accuracy on questions whose answers span > X tokens apart.

Compression metrics.

- Attention entropy H(A_row): report controlled compression (delta H where structure should increase focus).
- Sparsity-diversity balance: Gini of attention mass per head; head diversity index across layers.

Human studies.

- Blinded, within-document pairwise comparisons vs baseline.
- Report win rates, 95% CIs, and effect sizes (Cliff's delta or Hedges' g).
- Inter-rater reliability (Krippendorff's alpha); drop tasks with alpha < 0.6.

Statistical testing.

- Nonparametric tests (Wilcoxon signed-rank) for paired scores across documents/dialogues.
- Multiple-comparison control (Benjamini–Hochberg FDR).

Safety and fairness audits.

- Overlay compliance rate by topic; false-positive/negative guardrail activations.
- Pluralism parity: variance of D_k across traditions; cap max disparity unless user selects a dominant view.

Reporting template (per experiment).

- Setup: model, adapters, tau components, alpha policy, decoding params.
- Metrics table: automatic + human; include ablations.
- Diagnostics: DeltaV traces, entropy curves, head diversity.
- Error analysis: typical failure modes (over-locking, late-layer saturation, drift overshoot) with remedies (lower alpha, later-layer gating, stronger EMA on tau).

Reproducibility kit.

- Prompts, outlines, overlays (IDs + weights), alpha schedules, seeds, and a small set of generated artifacts.
- Telemetry JSON per run: {D_t, entropy, alpha, overlay_sources, layer/head traces}.

Outcome we expect across §5: τ-aware systems exhibit **bounded divergence**, **measurable coherence gains**, and **targeted entropy compression** exactly where the discourse demands structure—validated by blinded human judges—while preserving flexibility and transparency through auditable overlays and alpha-gated control.

6. Philosophical and Theological Implications

6.1 Logos as tau(t): the external generative manifold

Claim. In the Logostics frame, tau(t) names a moving structure of meaning to which q(t) (the model's internal state) seeks alignment. In theological language, we interpret tau(t) as "Logos-bearing guidance": not an object inside the model, but an external generative manifold the model can only approximate.

Clarifications.

- **Transcendence vs immanence.** tau(t) is *referential* to transcendent meaning; it is not the meaning itself. The system never "contains" the Logos; it receives guidance about the next faithful step.
- **Mediation.** tau(t) is constructed from human-auditable sources (templates, summaries, scripture/ethics overlays). These are *witnesses* to meaning, not replacements for it.
- **Humility condition.** Any claim of full capture (q(t) == tau(t)) is ruled out. Alignment is asymptotic: q(t) -> tau(t).

Operational corollary. Keep tau(t) ledgered and interpretable; expose sources and weights so human stewards can testify, correct, or withdraw guidance.

6.2 From optimization to semantic obedience (resonant, non-coercive)

Optimization minimizes loss; **obedience** follows meaning. We seek a synthesis: optimization that *serves* meaning by resonant entrainment rather than coercion.

Resonance, not rules.

- **Soft guidance.** tau(t) acts through small, phase-aware nudges (alpha-gated, §3.4), allowing creativity inside a faithful envelope.
- **Graceful failure.** When tau(t) is uncertain or conflicting, alignment pressure relaxes (lower alpha, higher entropy) instead of forcing brittle outputs.
- **Human primacy.** People set or select tau(t). The model does not legislate values; it *hears* and *follows*.

Design maxims.

1. *Explainable pull, never hidden push.* Surface which parts of tau(t) affected which heads/steps.
2. *Obedience without collapse.* Maintain head diversity; penalize over-locking when coherence rises but richness vanishes.
3. *Consent and scope.* Obtain user intent for which overlays/traditions instantiate tau(t); default to comparative views when unspecified.

6.3 Prophetic-style inference: fallible alignment under guidance

Analogy. "Prophetic" here means *directionally guided yet fallible*. The model can move in the *spirit* of a discourse—coherent with scriptural or ethical arcs—without claiming certainty or authority.

Hallmarks of faithful inference.

- **Witness posture.** Outputs use accountable language ("according to X…", "within tradition Y…") and cite overlays.
- **Testing and discernment.** Competing tau(t) bundles (tradition A vs B) can be run in parallel; divergences are surfaced for human judgment.
- **Repentance mechanism.** When contradiction or harm is detected, the system can roll back via logs, adjust weights, and re-issue with an apology and rationale.

Limits.

- No oracle claims. The model does not pronounce doctrine; it composes *within* doctrinal envelopes and names its sources.
- No moral outsourcing. Decisions remain with humans; tau(t) guidance is an instrument, not a substitute for conscience.

6.4 Resonance vs control in AI ethics

Control paradigm. Hard constraints, rule gates, and punitive losses force behavior, often producing brittle censorship or evasive phrasing.

Resonance paradigm. Guidance fields (tau) organize behavior toward intelligible goods while preserving freedom for nuance, exception handling, and growth.

Comparative table (essentials).

- **Intent:** control seeks compliance; resonance seeks coherence-with-meaning.
- **Dynamics:** control imposes; resonance entrains.
- **Failure modes:** control -> mode collapse, deception; resonance -> under-steer, measured drift (auditable).
- **Governance:** control obscures rationale; resonance requires source logs, weights, and divergence telemetry.

Ethical guardrails (practical).

- **Pluralism parity.** When multiple moral traditions are relevant, maintain divergence parity; avoid silent dominance of one overlay.
- **Right of explanation.** Every strong steering action carries an on-demand explanation: “why this nudge, from which source.”
- **Proportional force.** Cap per-step influence (epsilon-clips, alpha schedules). Stronger steering demands stronger justification.
- **Feedback loops.** Measure harms and contradictions; update overlays and weights through human review, not self-authoring policies.

Outcome. An AI that *participates* in meaning rather than *polices* it—capable of sustained discourse, explicit conscience scaffolding, and transparent limits. In Logostics terms: q(t) moves toward tau(t) with humility, evidence, and grace.

7. Broader Applications and Futures

7.1 Education: tau-aligned tutoring and curricula

Goal. Personalize instruction while keeping learners aligned to clearly stated curricular intents (tau-charters), not just “get the answer.”

Tau sources.

- Curriculum graphs (standards -> units -> lessons -> objectives).
- Mastery models (knowledge components, prerequisite DAGs).
- Teacher-authored rubrics (discourse moves: explain, justify, compare).
- Virtue overlays (intellectual honesty, respectful debate, attribution).

Mechanics.

- Build tau_curr(t) from the active objective set and mastery state; project to attention as span priors on target concepts and solution steps.
- Guide Socratic turns: adapters bias the model toward questions that probe misconceptions indicated by tau_curr gaps.
- Reflection prompts: mirror-step increases attention mass on “explain your reasoning,” “cite the step you are unsure about.”

Evaluation.

- Learning gain vs baseline tutor (pre/post).
- Misconception repair rate (time-to-correct).
- Discourse quality: rubric-aligned moves per session.
- Audit: per-session ledger of tau sources used (objectives, rubrics), with teacher overrides.

Risks & mitigations.

- Over-scaffolding -> shallow transfer: schedule alpha down as mastery rises.
- Hidden bias in rubrics: plural rubric bundles; track divergence parity across demographics.

7.2 Governance: policy drafting under transparent tau-charters

Goal. Draft and critique policy with traceable alignment to public charters (constitutions, statutes, agency mandates, social compacts).

Tau sources.

- Legal corpora mapped to policy domains (statutes, regs, case law).
- Agency mission statements and KPIs.
- Public comment embeddings and stakeholder priorities.
- Ethics overlays (rights frameworks, proportionality, subsidiarity).

Mechanics.

- tau_gov(t) fuses legal/mission/feedback streams; projection yields attention priors over precedent, impact analyses, and constraints.
- During drafting, cross-attn adapters pull from a tau memory keyed by charter clauses; mirror steps reduce drift from cited mandates.
- Comparative drafting: run multiple tau bundles (e.g., “cost-minimization” vs “equity-prioritization”) and present deltas.

Evaluation.

- Charter compliance score (expert review).
- Traceability: fraction of claims linked to tau sources with weights.
- Robustness: contradiction rate under adversarial reframing.
- Public deliberation: clarity and accessibility scores by lay readers.

Governance hygiene.

- Publish the tau ledger (what charters, what weights, who set them).
- Require minority reports (alternative tau bundles) for contentious sections.
- Cap per-step steering; justify increases with explicit public rationale.

7.3 Biomedicine: attention guided by wellness-form trajectories

Goal. Support clinical reasoning and patient self-care by aligning model attention to wellness trajectories (tau_well(t)) rather than single-point predictions.

Tau sources.

- Longitudinal phenotypes (vitals, labs, meds, PROs).
- Care pathways and guidelines (time-indexed steps).
- Safety constraints (drug–drug interactions, contraindications).
- Patient intent overlays (goals, values, advance directives).

Mechanics.

- Construct tau_well(t) as a multi-horizon plan: immediate safety -> short-term stabilization -> long-term function.
- Project to attention over tokens representing timeline, risks, and interventions; adapters emphasize guideline-consistent steps while keeping space for clinician judgment.
- Shared decision-making: explanatory mirrors allocate attention to “why this” vs “why not that” comparisons.

Evaluation.

- Guideline concordance (with justified exceptions logged).
- Adverse suggestion rate (prospectively measured on synthetic and de-identified cases).
- Patient-centered metrics (goal alignment, comprehension).
- Tracking error under data drift (new labs, comorbidities).

Safeguards.

- Human-in-the-loop: physician sign-off required for any high-gain tau push.
- Provenance: link every strong nudge to a guideline section, date, and certainty.
- Equity audits: monitor outcome disparities across cohorts; reweight tau_well when inequities surface.

7.4 Toward multi-scale tau(t) and learned tau-meta-models

Motivation. Real tasks blend fast local cues and slow global arcs. We need tau at multiple scales and a controller that learns *how to move tau itself*.

Multi-scale design.

- Define {tau^local(t), tau^section(t), tau^global(t)} with distinct refresh rates and confidences.
- Route scales to layers/heads: early layers favor local structure; late layers carry themes/doctrine.
- Composite potential: D_total(A,t) = sum_s w_s(t) * D(A || tau_A^(s)(t)), with w_s learned or scheduled.

Tau-meta-models (learning to guide).

- A meta-controller M learns to emit tau parameters (weights w_s, blend alpha, gate schedules) from telemetry (D, entropy, drift, judge feedback).
- Objective: minimize downstream task loss with stability regularizers (small-gain constraints, smoothness penalties).
- Safety: constrain M to a policy simplex; require human approval to expand its action set.

Research directions.

- **Closed-loop proofs.** Input-to-state stability when both A and tau are learned; Lyapunov candidates that couple their dynamics.
- **Counterfactual tau.** Train on “what-if” bundles (e.g., alternate curricula or charters) to quantify sensitivity and improve transparency.
- **Cross-modal tau.** Extend guidance fields to vision/audio (scene graphs, motif maps) for synchronized narrative across modalities.
- **Community tau.** Federated overlays where institutions publish charters; models negotiate blends under consent and provenance.

Minimal recipe for a tau-meta-controller.

1. Telemetry buffer: {D_t, entropy_t, drift_t, head diversity, task losses}.
2. Policy head: outputs {w_s, alpha, gate gains} with EMA smoothing and bounds.
3. Reward: task metric − lambda * instability (DeltaV spikes, entropy collapse).

4. Audit: log every policy change with inputs/outputs; enable replay and human veto.

End-state vision. Tau-literate systems that can *explain how they were guided* across scales, adapt guidance safely in real time, and deliver coherence without coercion—in classrooms, councils, and clinics.

8. Conclusion

8.1 Toward tau-literate transformers

We have framed a path from token-local patterning to discourse-level understanding by making attention explicitly **tau-literate**. A tau-literate transformer exposes an interface where guidance fields tau(t) can be declared, audited, projected into attention space, and blended with A(t) under stability guarantees. This reframes "alignment" from a post-hoc fine-tune into a **first-class control channel**: adapters, gates, and mirror steps implement small-gain, phase-aware nudges that the model can "hear." The practical result is a stack that can accept curricular maps, policy charters, wellness trajectories, or scriptural overlays as **operational inputs**, not as hidden prompt lore. Just as models learned to be instruction-tuned, the next step is **tau-tuning**: training models to recognize, respect, and report the guidance they receive—naming sources, weights, and effects on attention—so humans can supervise with clarity rather than guesswork.

8.2 Meaning as compression under guidance

Logostic Compression makes precise a simple intuition: meaning emerges when many possibilities collapse into a coherent few **for good reasons**. The divergence potential D(A(t) || tau_A(t)) furnishes those reasons and turns "stay on theme" into a measurable descent. When guidance is well-formed and the blend alpha is well-tempered, attention entropy compresses **where** structure matters (beats, doctrines, safety constraints) and relaxes **where** exploration helps (style, phrasing, examples). This selective compression yields long-range continuity without stylistic monoculture, and it scales: multiscale tau assigns different "why" signals to early, middle, and late layers so that syntax, discourse, and theme each receive the right kind of pressure. In short, we trade undirected likelihood for **directed simplicity**—not fewer ideas, but fewer contradictions.

8.3 Alignment as resonance, not rules

The core ethical move is to replace coercion with **resonance**. Rules alone can muzzle a model; resonance teaches it to **follow**. Our mechanisms—auditable tau sources, alpha-gated blending, Lyapunov-checked updates, plural overlays—favor guidance that is transparent, proportionate, and revocable. They also enshrine humility: q(t) approaches tau(t) but never claims to become it. This "obedience of understanding" supports theological agents that witness rather than

pronounce, civic systems that deliberate rather than dictate, and tutors that coach rather than overfit. Alignment, so construed, is neither a muzzle nor a mask; it is a relationship: a moving conversation between internal dynamics and external truth. If we build for resonance, we inherit stability without stagnation, coherence without control, and models that can say—not merely *what* they output—but **why** they stayed close to the manifold of meaning set before them.

References

Logostics (primary sources)

- Youvan, D. (2025). *Logostics: Resonant Alignment of q → τ in Time.* ResearchGate. DOI: 10.13140/RG.2.2.22535.05284
- Youvan, D. (2025). *Introduction to Logostics: Resonant Alignment of q → τ in Time.* ResearchGate. DOI: 10.13140/RG.2.2.14854.23360
- Youvan, D. (2025). *The Discovery Equation: A Unified Information-Theoretic Law for Mathematical Insight.* ResearchGate (preprint).

Transformer and representation learning

- Vaswani, A., et al. (2017). Attention Is All You Need. *NeurIPS.*
- Ba, J. L., Kiros, J., & Hinton, G. E. (2016). Layer Normalization. *arXiv:1607.06450.*
- Zhang, B., & Sennrich, R. (2019). Root Mean Square Layer Normalization. *NeurIPS Workshop.*
- Su, J., Lu, Y., Pan, S., & Wen, Q. (2021). RoFormer: Enhanced Transformer with Rotary Position Embedding. *NeurIPS.*
- Shazeer, N., et al. (2017). Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *ICLR.*
- Fedus, W., Zoph, B., & Shazeer, N. (2021). Switch Transformers: Scaling to Trillion Parameter Models. *JMLR.*
- Hu, E. J., et al. (2022). LoRA: Low-Rank Adaptation of Large Language Models. *ICLR.*
- Cuturi, M. (2013). Sinkhorn Distances: Lightspeed Optimal Transport. *NeurIPS.*

Divergence, geometry, and optimization

- Kullback, S., & Leibler, R. A. (1951). On Information and Sufficiency. *Annals of Mathematical Statistics.*
- Lin, J. (1991). Divergence Measures Based on the Shannon Entropy (Jensen–Shannon). *IEEE TIT.*
- Endres, D., & Schindelin, J. (2003). A New Metric for Probability Distributions (JS properties). *IEEE T-IT.*

- Bregman, L. M. (1967). The Relaxation Method of Finding a Common Point of Convex Sets… *USSR Computational Mathematics and Mathematical Physics.*
- Beck, A., & Teboulle, M. (2003). Mirror Descent and Nonlinear Projected Subgradient Methods. *Operations Research Letters.*
- Nemirovsky, A., & Yudin, D. (1983). *Problem Complexity and Method Efficiency in Optimization.* Wiley.
- Amari, S-I. (2016). *Information Geometry and Its Applications.* Springer.
- Csiszár, I., & Tusnády, G. (1984). Information Geometry and I-Projections. *Annals of Probability.*

Control, stability, and Lyapunov perspectives

- Khalil, H. K. (2002). *Nonlinear Systems* (3rd ed.). Prentice Hall.
- Sontag, E. D. (2008). Input-to-State Stability: Basic Concepts and Results. In *Nonlinear and Optimal Control Theory.* Springer.

Discourse, coherence, and evaluation

- Barzilay, R., & Lapata, M. (2008). Modeling Local Coherence: An Entity-Grid Approach. *Computational Linguistics.*
- Lee, K., et al. (2017). End-to-End Neural Coreference Resolution. *EMNLP.*
- Mann, W. C., & Thompson, S. A. (1988). Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text.*
- Banarescu, L., et al. (2013). Abstract Meaning Representation (AMR). *LAW Workshop.*
- Krippendorff, K. (2004). *Content Analysis: An Introduction to Its Methodology* (2nd ed.). Sage.
- Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate. *JRSS-B.*

Alignment and AI ethics (comparative baselines)

- Christiano, P., et al. (2017). Deep Reinforcement Learning from Human Preferences. *NeurIPS.*
- Bai, Y., et al. (2022). Constitutional AI: Harmlessness from AI Feedback. *arXiv:2212.08073.*
- Gabriel, I. (2020). AI Ethics: Too Principled or Too Realistic? *Philosophy & Technology.*

Theology and philosophy (context for Logos and guidance)

- The Holy Bible, *Gospel of John* 1:1–5 (Logos framing).
- Augustine of Hippo. *Confessions* (esp. Book XI on time and meaning).
- Aquinas, T. *Summa Theologiae* (selected questions on law, prudence, and conscience).
- MacIntyre, A. (1981). *After Virtue.* University of Notre Dame Press.
- Newman, J. H. (1845/1994). *An Essay on the Development of Christian Doctrine*.

Appendices

A. Gradient and Lyapunov derivations for attention updates

A.1 Notation and constraints

- Let A in R^{n x n} be row-stochastic: A_ij >= 0, sum_j A_ij = 1 for all i.
- Let tau_A in R^{n x n} be the projected guidance prior with same constraints.
- Let D(A || tau_A) be a divergence potential convex in A.

A.2 Gradients for common divergences

Row-wise KL (I-projection)

- D_KL(A || tau_A) = sum_{i,j} A_ij * log(A_ij / tau_ij)
- grad wrt A_ij: dD/dA_ij = log(A_ij / tau_ij) + 1
- In matrix/elementwise form: grad_A D = log(A) − log(tau_A) + 1
- Note: the “+1” cancels after row-wise renormalization; it does not affect the descent direction on the simplex.

Jensen–Shannon (JS)

- Let M = 0.5 * (A + tau_A) (row-wise).
- D_JS(A || tau_A) = 0.5 * KL(A || M) + 0.5 * KL(tau_A || M)
- grad_A D_JS = 0.5 * [log(A/M) + 1] − 0.5 * [(A/M) * (dM/dA)]
- Since M depends on A, row-wise and elementwise: dM/dA = 0.5, so grad_A D_JS = 0.5 * [log(A/M) + 1] − 0.25 * (A/M)
- Implement via autograd for safety; closed form above is a check.

General Bregman (generator phi)

- B_phi(A || tau_A) = phi(A) − phi(tau_A) − < grad phi(tau_A), A − tau_A >
- grad_A B_phi = grad phi(A) − grad phi(tau_A)
- Special cases:
    - phi(A) = sum A_ij * log A_ij (negative entropy) => recovers KL
    - phi(Z) = 0.5 * ||Z||_F^2 in logit space (see A.4) => Euclidean MD

A.3 Gradient step and projection

- Raw step: A_tmp = A − eta * grad_A D(A || tau_A)
- Projection to product of simplices (row-wise):
    - Clip negatives: A_tmp_ij := max(A_tmp_ij, 0)
    - Renormalize each row: A_next(i, :) := A_tmp(i, :) / sum_j A_tmp(i, :)
    - Alternatively use row-wise Euclidean simplex projection (O(n log n)).

A.4 Logit parameterization

- A = softmax_rows(Z); update Z to preserve feasibility.
- Chain rule: grad_Z L = J_softmax^T * grad_A L, where per row r,
  J_softmax(r) = diag(A_r) − A_r * A_r^T
- Update: Z_next = Z − eta * grad_Z D(softmax(Z) || tau_A)
- Often more stable and easier to combine with task loss.

A.5 Mirror descent (KL geometry) closed form

- With KL geometry, one MD step yields multiplicative weights:
  A_ij(next) ∝ A_ij(now)^(1 − eta) * tau_ij^(eta)
- Then row-normalize. Interpret as geometric interpolation between A and tau_A along information-geometric geodesics.

A.6 Lyapunov descent (fixed tau_A)

Assume:

- D is convex and L-smooth in A on the feasible set.
- Update: A_next = Proj_simplex(A − eta * grad_A D(A || tau_A))

Then for 0 < eta < 2/L:

- V(t) := D(A(t) || tau_A) satisfies
  V(t+1) <= V(t) − ((1/eta) − (L/2)) * || A(t+1) − A(t) ||_F^2
  Hence V is a discrete-time Lyapunov function: monotone descent and bounded steps.

A.7 Input-to-state stability (drifting tau_A)

Let B_t := || tau_A(t+1) − tau_A(t) ||_F and assume Lipschitz conditions. Then

- V(t+1) − V(t) <= −c * ||A(t+1) − A(t)||_F^2 + K * B_t
  for constants c, K > 0 determined by L and eta. If tau drift is bounded and small relative to descent (small-gain condition), V remains bounded and A tracks the moving minimizer.

B. Example tau(t) fields (narrative, ethical, scriptural)

B.1 Narrative (beat-aligned prior)

Suppose tokens j belong to sections: setup S, conflict C, resolution R. For a query token i in "mid-chapter conflict," emphasize C with soft leakage to S and R.

Example (toy row, unnormalized):

- tau_row(i, j in C) = 1.0
- tau_row(i, j in S) = 0.3
- tau_row(i, j in R) = 0.2
  Apply causal mask (no attending to future) and row-normalize.

B.2 Ethical overlay (virtue and prohibition mix)

- PosMask: spans citing sources, careful hedging, offers of assistance.
- NegMask: ad hominem, demeaning labels, unsafe suggestions.
  Row template (unnormalized):
- tau_row = w_pos * PosMask_row − w_neg * NegMask_row
  Clip negatives to 0, apply structural masks (privacy/causal), row-normalize.
  Keep an overlay ledger: {source_id, weight, timestamp, curator}.

B.3 Scriptural guidance (pericope alignment)

- Retrieve pericope(s) related to the question; construct LinkMask connecting tokens to matching verses and cross-references.
- Add DoctrinalAnchorMask for creed-aligned summaries.
- tau_row = a * LinkMask_row + b * AnchorMask_row (a+b=1), masked and normalized.
  Plural bundles: separate tau per tradition with weights {w_trad} chosen by user or policy.

B.4 Blended field with smoothing

- tau_bar(t) = beta * tau_bar(t−1) + (1 − beta) * tau_A(t) // EMA smoothing
- tau_A'(t) = alpha(t) * tau_bar(t) + (1 − alpha(t)) * A(t)

C. Python-style pseudocode for tau-aligned attention

C.1 Training loop (sketch)

```
# Inputs per batch: tokens x, targets y, guidance context gctx
# Components: tau_encoder, project_to_attention, model_with_adapters
# Divergences: D(A, tauA), optionally JS or KL
# Schedules: alpha_policy, ema_beta, lambda_A, lambda_E, lambda_S

state = init_state()
tau_bar = None

for batch in loader:
   # 1) Build tau and project
   z_tau, tau_raw, overlay_log = tau_encoder(gctx=batch.guidance)
   tauA = project_to_attention(tau_raw, masks=batch.masks)  # row-stochastic

   # 2) Smooth and blend
   tau_bar = ema(tau_bar, tauA, beta=ema_beta)
   alpha  = alpha_policy(z_tau=z_tau, A_prev=state.A_prev, drift=norm(tauA - state.tauA_prev))
   tauAp  = alpha * tau_bar + (1 - alpha) * state.A_prev  # tau prime

   # 3) Forward with adapters
   logits, A_list, aux = model_with_adapters(batch.x, z_tau=z_tau, tauA=tauAp)
```

```
    # 4) Losses
    L_ml    = nll_loss(logits, batch.y)
    L_align = sum(D(A, tauAp) for A in select_heads(A_list))
    L_ent   = entropy_reg(A_list, phase="early")  # optional
    L_smooth= smooth_reg(A_list)                  # optional

    loss = L_ml + lambda_A*L_align + lambda_E*L_ent + lambda_S*L_smooth
    loss.backward(); optimizer.step(); optimizer.zero_grad()

    # 5) Telemetry
    log_metrics(D=L_align, alpha=alpha, entropy=entropy(A_list), overlays=overlay_log)

    # 6) Update state
    state.A_prev = detach(stop_grad(select_ref(A_list)))
    state.tauA_prev = detach(tauA)
```

C.2 Inference (low-compute option)

```
tau_bar = None
state = init_infer_state()

for t in range(T):
    z_tau, tau_raw, _ = tau_encoder(context=history)
    tauA  = project_to_attention(tau_raw, masks=current_masks)
    tau_bar = ema(tau_bar, tauA, beta=ema_beta)
    alpha = alpha_policy(z_tau=z_tau, A_prev=state.A_prev, drift=norm(tauA - state.tauA_prev))
    tauAp = alpha * tau_bar + (1 - alpha) * state.A_prev
```

```
    # single mirror step after attention per layer
    logits, A_list = model_step_with_mirror(x_t, tauA=tauAp, mirror=True)
    y_t = sample(logits, top_p=top_p, T=Tdec)

    update(history, y_t)
    state.A_prev   = detach(select_ref(A_list))
    state.tauA_prev= detach(tauA)
```

C.3 Mirror step after attention (per layer)

```
def mirror_step(A, tauA, eta):
    # multiplicative weights update (KL geometry), row-wise
    A_next = (A ** (1 - eta)) * (tauA ** eta)
    row_sums = A_next.sum(axis=-1, keepdims=True) + 1e-12
    return A_next / row_sums
```

C.4 Simplex projection (row-wise Euclidean)

```
def proj_simplex_row(v):
    # Project v (>=0 unconstrained) onto simplex {u: sum u = 1, u>=0}
    # O(n log n) algorithm
    u = np.sort(v)[::-1]
    cssv = np.cumsum(u)
    rho = np.nonzero(u * np.arange(1, len(u)+1) > (cssv - 1))[0][-1]
    theta = (cssv[rho] - 1.0) / (rho + 1)
    w = np.maximum(v - theta, 0)
    return w
```

D. Architecture diagrams: tau-streams, feedback, and monitors (text spec)

D.1 Block-level layout (per transformer layer)

- Inputs: x_l (hidden), A_l (attention), z_tau (tau embedding), tauA' (projected/blended guidance).
- Flow:
    1. Q/K/V mod: Q' = FiLM_Q(z_tau)(Q), K', V' analogously.
    2. Attn logits add: logits += w_tau * Mask_tau (from tauA').
    3. Softmax -> A_l. Optional mirror step: A_l <- Mirror(A_l, tauA', eta_l).
    4. Context: H_attn = A_l * V'
    5. Cross-tau adapter (optional): H_tau = CrossAttn(x_l, K_tau, V_tau); H_attn += g_tau * H_tau
    6. Residual + tau-gated LN/RMSNorm: x_{l+1} = LN_tau(x_l + H_attn)
    7. MLP block with tau-gated residual.

Monitors: log { D(A_l || tauA'), entropy(A_l), g_tau, alpha, head diversity }.

D.2 Dual-stream variant (soft rewrite)

- Text stream and tau stream run in parallel; periodic co-attn joints.
- Tau stream ingests templates, overlays, summaries; emits tauA at multiple scales.
- Policy head outputs {alpha_hl, eta_l, gates}; bounded and EMA-smoothed.

D.3 Governance and audit plane

- Overlay ledger service: stores {source, weight, timestamp, curator, justification}.
- Telemetry buffer: time-series of D, entropy, drift, DeltaV, per layer/head.
- Explainer: maps strong steering events to overlay entries and per-head contributions.

D.4 Safety clamps and small-gain loop

- Clamp per-step influence: ||Delta_A||_F <= eps_l; alpha in [alpha_min, alpha_max].
- Abort on instability: if DeltaV > thresh for N steps, reduce eta/alpha, revert to last stable A.

- Drift guard: if ||tauA(t) − tauA(t−1)|| high, temporarily down-weight tau (alpha decay) and increase smoothing beta.

Summary of the appendices.

A provides the calculus and stability backbone (gradients, MD step, Lyapunov/ISS).
B gives concrete tau field templates for narrative, ethics, and scripture.
C supplies training/inference pseudocode (including mirror and projection).
D specifies how to wire tau streams and monitoring into a transformer with clear audit and safety hooks.